**WhatsApp NET Client**

**WhatsApp Client Demo App for .NET Developer**

**What is WhatsApp NET Client?**

[WhatsApp NET Client](http://wa-net.coding4ever.net/) is a **free** demo app for .NET Developers which is used to develop desktop based WhatsApp client applications. This library was developed on .NET technology using the C# programming language, so it can also be used for all .NET programming languages ​​other than C# such as VB.NET, F#, C++ and other .NET languages.

In its development, WhatsApp NET Client uses [whatsapp-web.js](https://github.com/pedroslopez/whatsapp-web.js/) as an engine to communicate with WhatsApp Web.

So in conclusion, by using this sample you can easily create applications that require communication with the WhatsApp Web application, such as:

* WhatsApp Blast
* [WhatsApp Bot](https://github.com/k4m4r82/OnlineTestWABot)
* And other client WhatsApp Apps

**Feature**

* Full *stealth* mode (headless/no window). You can say goodbye to the chrome/firefox browser which usually appears to run WhatsApp Web.
* Automatically saves login sessions (so scan the WA qr code just once)
* Supports the use of [multiple WA accounts](https://github.com/WhatsAppNETClient/WhatsAppNETClientMultiAccount) , for examples you can see at <https://github.com/WhatsAppNETClient/WhatsAppNETClientMultiAccount>
* Grab contacts to read WA contacts so that the results can be saved to the database
* Grab groups and members to read group data and its members so that the results can also be saved to the database
* Send personal or group messages
* Send multiple messages (broadcast)
* Send messages with images, audio, video and all kinds of documents
* Send messages with images, audio, video and all kinds of documents via URL
* Send messages by type location
* ReplyMessage (quoted message)
* Can store images, audio, video, all types of documents including vcards from incoming messages (you can specify the storage location yourself)
* Can also read messages with type vcardandlocation
* Can read messages from groups and get sender information
* Subscribe to ChangeState events to monitor connection status changes
* Subscribe to incoming messages, so you don't need to add another timer object to read incoming messages
* Subscribe to messages sent, with this feature we can check whether the message sent was successful or failed
* Archive chat
* Delete chat
* Logout
* Can be easily integrated with all types of databases

**System Requirements**

* Windows 8, 10 and latest versions of windows
* .NET Framework 4.5 and latest version of .NET
* Node.js version 14.16.x or later
* [Software git](https://git-scm.com/downloads) (version control)

**Installation**

**1. Installing Node.js**

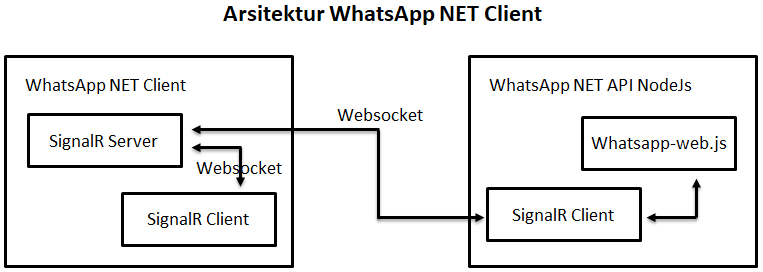
Why should I install Node.js? Because in the development of this WhatsApp NET Client, I used [whatsapp-web.js](https://github.com/pedroslopez/whatsapp-web.js/) as an engine to communicate with WhatsApp Web, so Node.js was needed because [Whatsapp-web.js](https://github.com/pedroslopez/whatsapp-web.js/) itself was developed on this platform.

To install Node.js is very easy, you just need to download the installation file [here](https://nodejs.org/en/download/) by choosing the version that suits my guide above. After that, just do the installation and follow the instructions.

**2. Mengclone Repository WhatsAppNETAPINodeJs**

[WhatsApp NET API NodeJs](https://github.com/WhatsAppNETClient/WhatsAppNETAPINodeJs) is a small program that I wrote using the java script programming language in which there is a module to access the [Whatsapp-web.js](https://github.com/pedroslopez/whatsapp-web.js/) and [node-signalr libraries](https://www.npmjs.com/package/node-signalr) .

* [Whatsapp-web.js](https://github.com/pedroslopez/whatsapp-web.js/) works to communicate with WhatsApp Web.
* [Node-signalr](https://www.npmjs.com/package/node-signalr) serves as a signalR client that is responsible for communicating with the signalR server that I planted in the WhatsApp NET Client app.

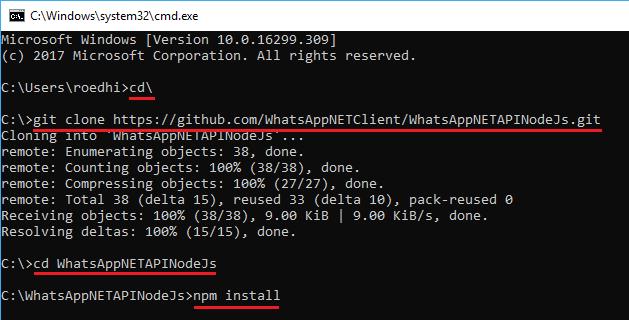


To clone the [WhatsApp NET API NodeJs repository,](https://github.com/WhatsAppNETClient/WhatsAppNETAPINodeJs) it's very easy, you just need to download the [git software](https://git-scm.com/downloads) and install it.

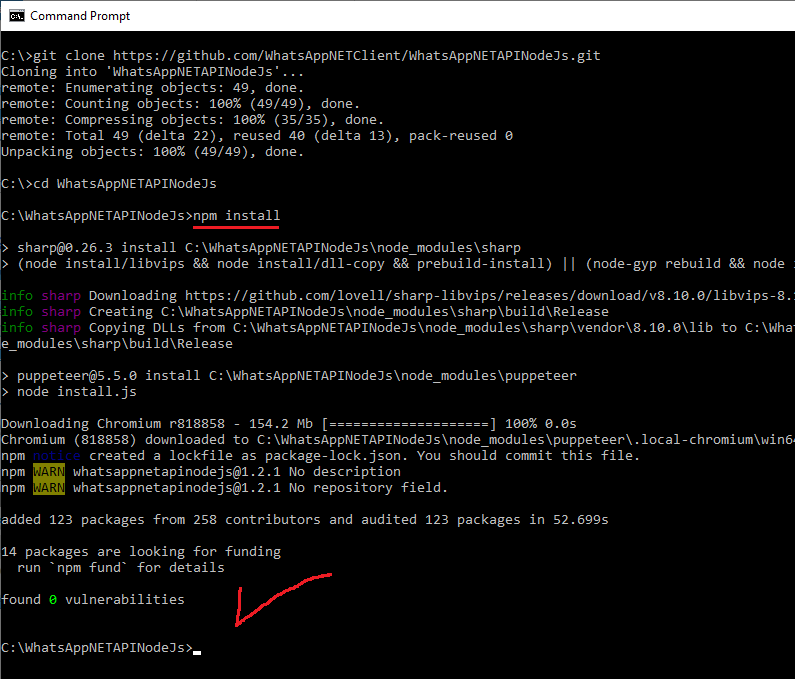
After that you run command promptthen go to drive Cor D, then type the following command:

git clone https://github.com/WhatsAppNETClient/WhatsAppNETAPINodeJs.git

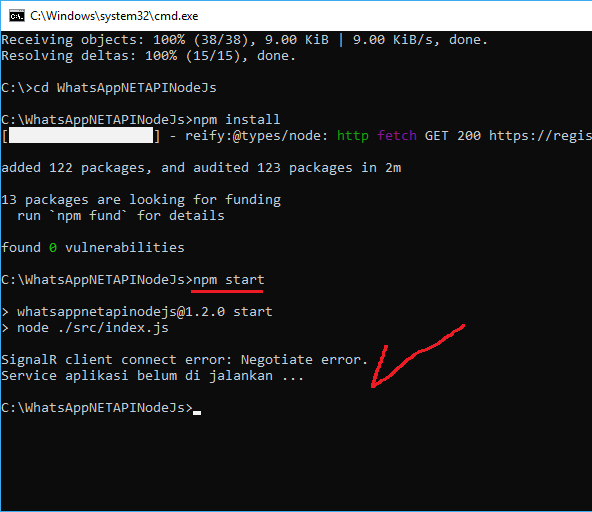
Don't forget to press enter, hehehe, after that continue by typing the command cd WhatsAppNETAPINodeJs, then type the last command, npm installand once again don't forget to enter.



Then wait until the installation of the required Node.js modules is complete. As for the display, the process is more or less like the following, so maybe on your pc/laptop you will get a different look.



Then how do you do if you want to test the installation is successful or not? It's easy, you just type the command npm startif the result is more or less like the picture below, I assume the installation has been successful :-).



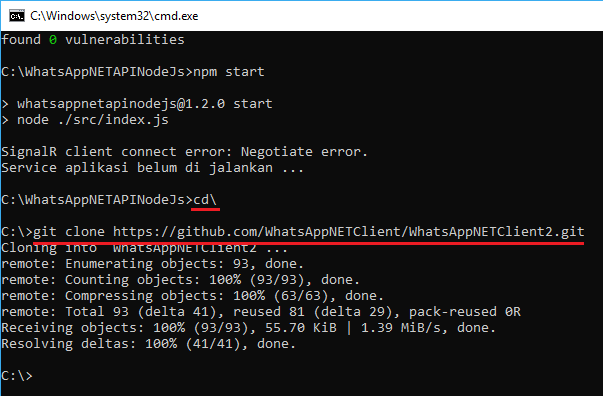
Let's just move on to the next step :-)

**3. Clone the Repository Example of Using the WhatsApp NET Client**

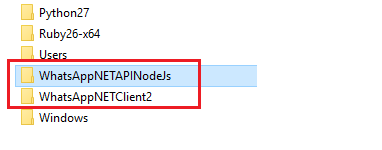
These steps are actually the same as before, you just have to change directories or want to move to another drive, it's the same. But for the steps here I'm still using the drive C to clone the repository for [examples of using the WhatsApp NET Client library](https://github.com/WhatsAppNETClient/WhatsAppNETClient2) .

Because our position is now in the directory C:\WhatsAppNETAPINodeJs, so we must first return to the root of the C drive by typing the command cd\, then type the following command:

git clone https://github.com/WhatsAppNETClient/WhatsAppNETClient2.git

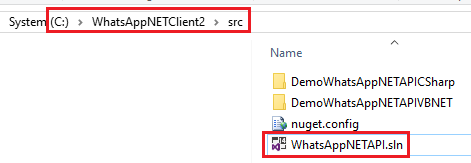


Ok, up here means the installation steps have been completed, if you open windows explorer then move to drive C, it will get two new directories with the name WhatsAppNETAPINodeJsand WhatsAppNETClient2which is the result of the previous steps.



**Opening the WhatsApp NET Client Usage Example Project**

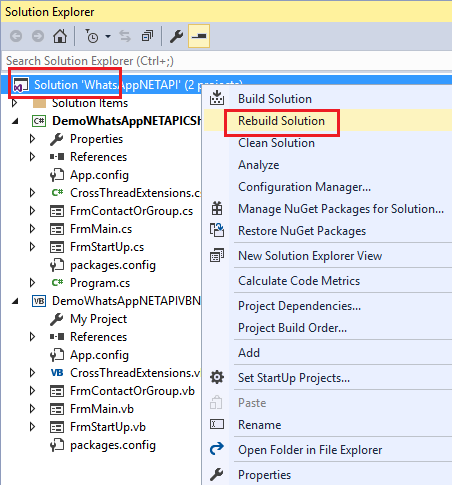
It's easy, just open the file WhatsAppNETAPI.sln in the C:\WhatsAppNETClient2\src.



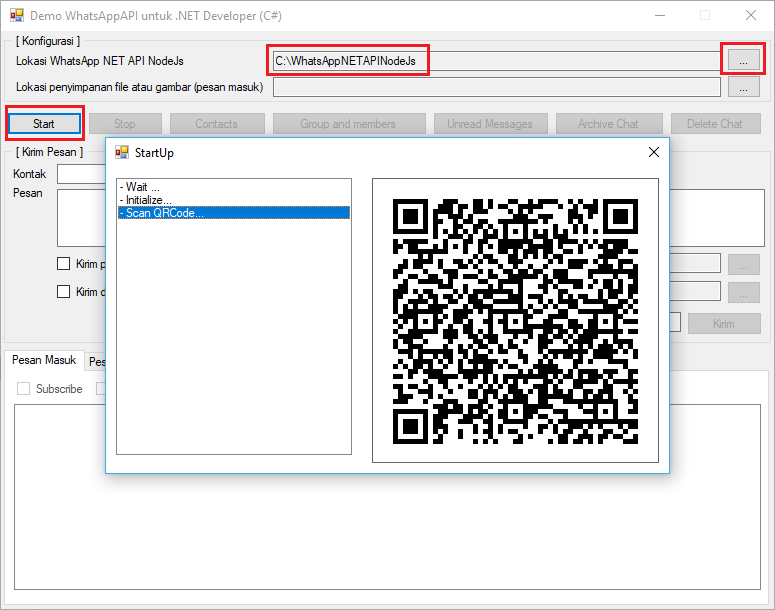
An example of using the WhatsApp NET Client library I created in two languages, namely C# and VB.NET.

Oh yes, while I remember hehe, I created this project using Visual Studio .NET 2015 Community, so at least you must have installed Visual Studio .NET 2015 Community or a newer version.

Ok after opening the WhatsAppNETAPI.sln, right click solution WhatsAppNETAPI-> Rebuild Solution. This step aims to restore all [nuget packages](https://www.nuget.org/packages/WhatsAppNETAPI) used in this project.



After that you can immediately run the project by pressing the F5 key, then set the directory location WhatsAppNETAPINodeJs which is the result of the previous steps, then click the *Start* button.



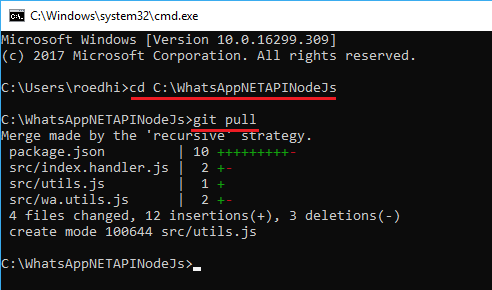
**Update**

**1. Updating the WhatsAppNETAPINodeJs Repository**

To update the [WhatsApp NET API NodeJs repository](https://github.com/WhatsAppNETClient/WhatsAppNETAPINodeJs) it's very easy, you run it first command prompt and then enter the directory WhatsAppNETAPINodeJs using the command CD, after that type the following command:

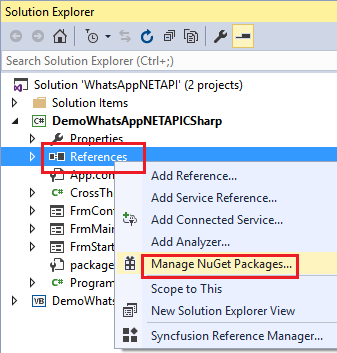
git pull

Then wait until the update process is complete. If successful, the process looks more or less like the following:

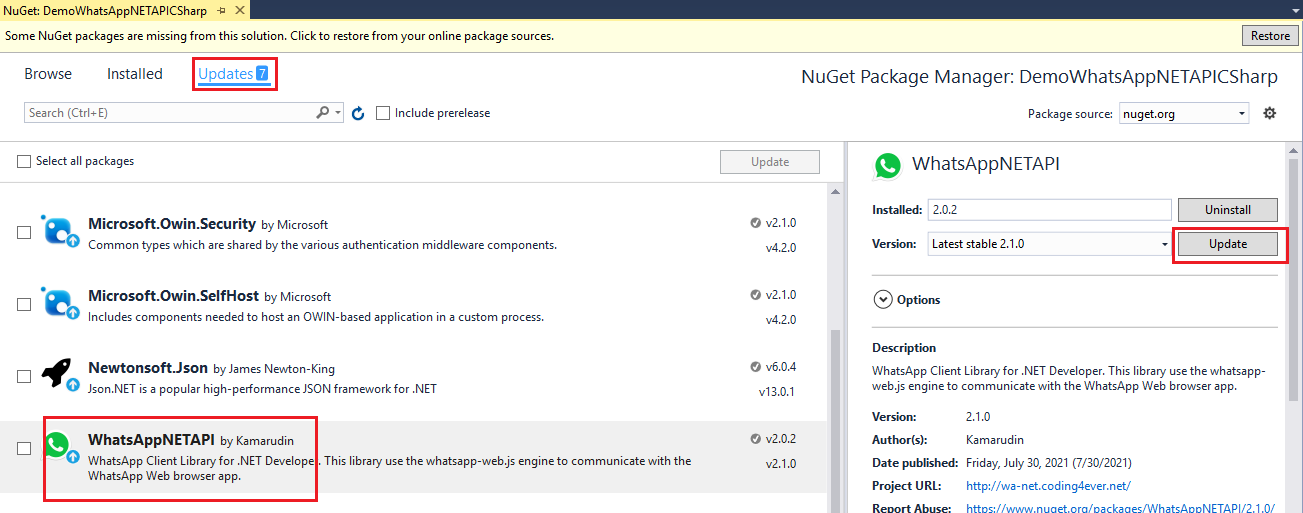


**2. Mengupdate Project .NET**

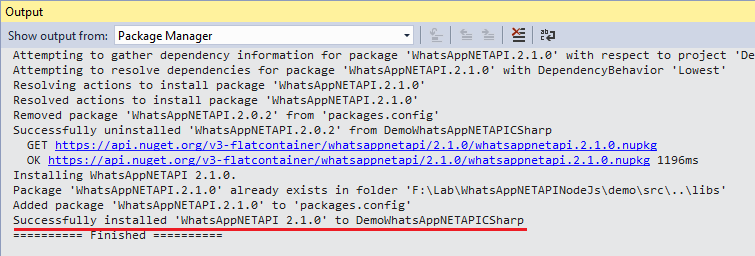
The next step is to update the [WhatsAppNETAPI](https://www.nuget.org/packages/WhatsAppNETAPI/) package nugget in your .NET project. The trick is to first open your project, then right click References->Manage Nuget Package...



Then in the Package Nugget window, you select the panel Updates, and find the [WhatsAppNETAPI](https://www.nuget.org/packages/WhatsAppNETAPI/) package then click the button Update.



Wait until the update process is complete. It looks more or less like this:

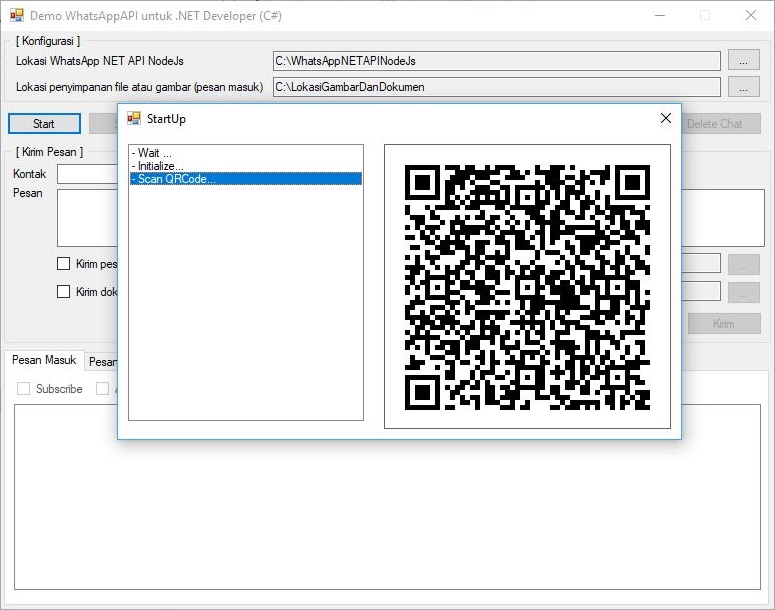


**Reporting a Bug or Error**

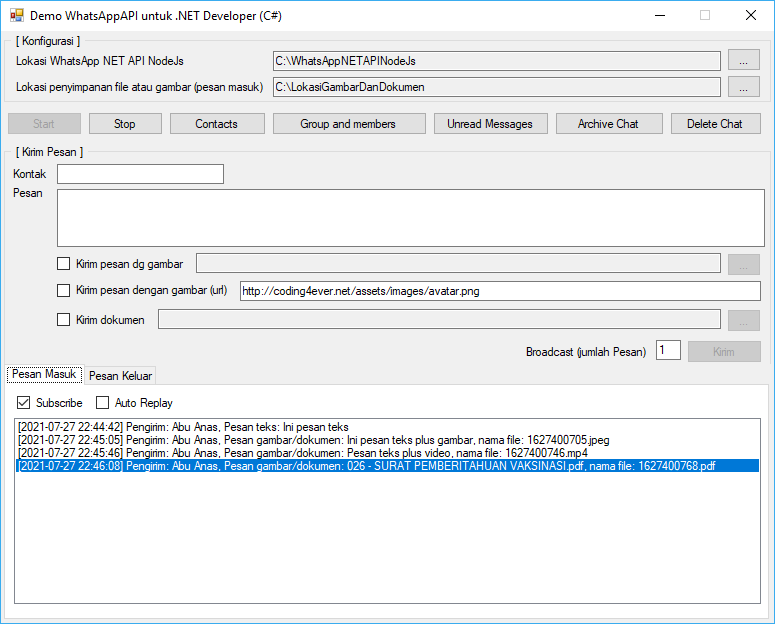
Technically in the development of an application it is clearly impossible to be 100% free of bugs. So if you find a bug or error when using the WhatsApp NET Client library, please report it on the <https://github.com/WhatsAppNETClient/WhatsAppNETClient2/issues> page .

**Project Screenshot Example of Using WhatsApp NET Client Library**

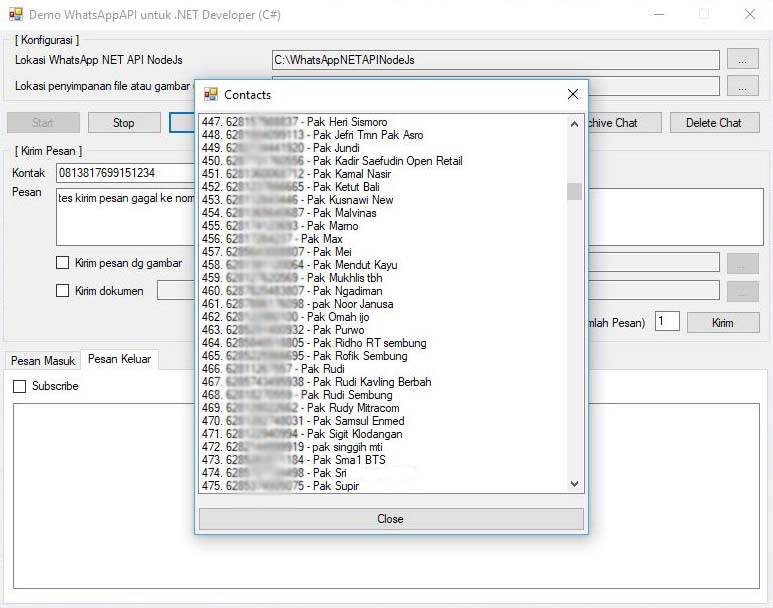
The initial scan of the QRCode is enough, for the next connection it is no longer needed because this library automatically saves the authentication session.



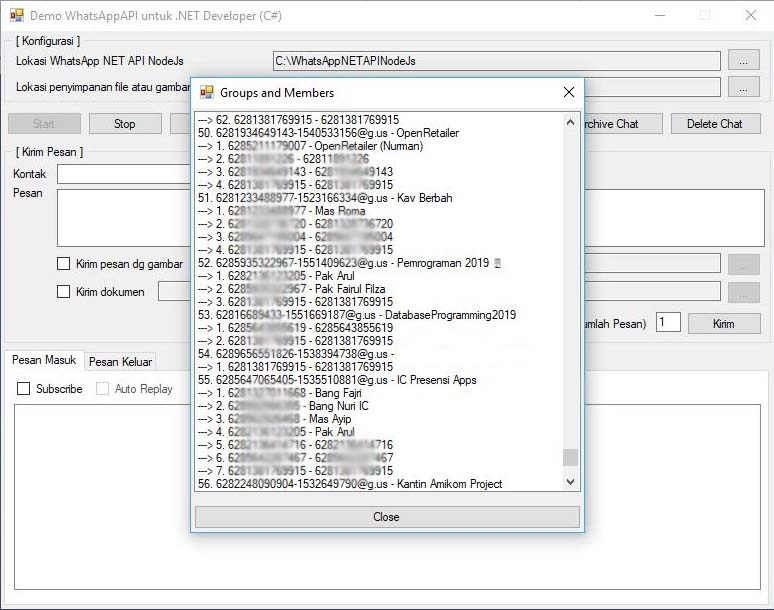
For example, subscribing to an incoming message, there is some information you can get such as date/time, sender's number/name, message and file name if the message contains images, audio, video and other document files.



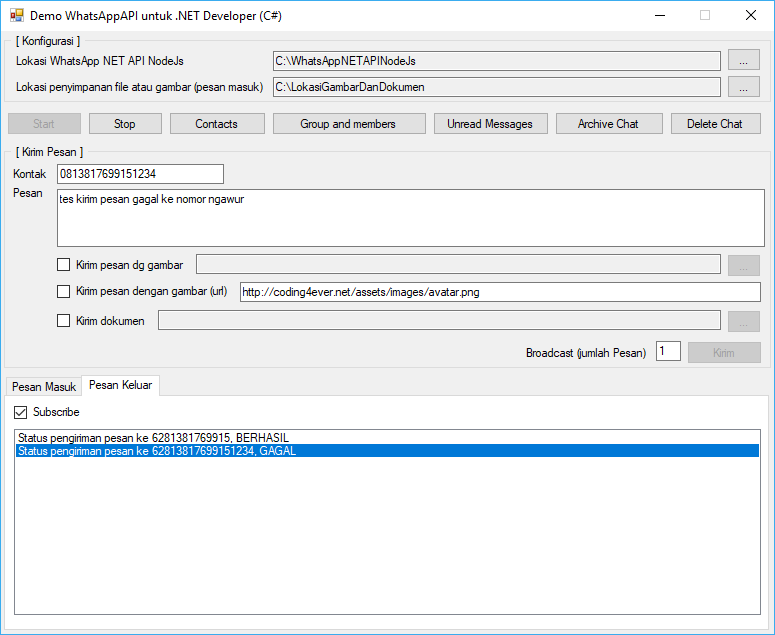
An example of the results of the contact data grab feature, in this feature you can get contact number/name information, short name and verified name.



An example of the results of the grab group feature and its members, in this feature you can get information on the name of the group and its members (contact number/name, short name and verified name).



An example of subscribing an outgoing message, in this feature you can get the status of sending a message whether it was successful or failed (the contact number has not been registered on WhatsApp).



**Release Info**

**v2.3.0 (2021-08-19)**

* New features:
  + Added [multi-account WA](https://github.com/WhatsAppNETClient/WhatsAppNETClientMultiAccount) features

In version [2.3.0](https://www.nuget.org/packages/WhatsAppNETAPI/2.3.0) there is an additional parameter sessionIdin all existing event handlers, so you have to adjust it manually.

Examples of some event handlers to customize:

**private** **void** **OnStartupHandler**(**string** message)

**private** **void** **OnScanMeHandler**(**string** qrcodePath)

**private** **void** **OnReceiveMessageHandler**(WhatsAppNETAPI.Message message)

All you have to do is add a parameter of sessionIdtype stringlike this:

**private** **void** **OnStartupHandler**(**string** message, **string** sessionId)

**private** **void** **OnScanMeHandler**(**string** qrcodePath, **string** sessionId)

**private** **void** **OnReceiveMessageHandler**(WhatsAppNETAPI.Message message, **string** sessionId)

**v2.2.0 (2021-08-12)**

* New features:
  + Added a feature to get group information without a member
  + Added a feature to send location
  + Added logout feature
* Bug fixes:
  + Subscribe to the status of outgoing messages sent to the group
* Improve:
  + Reading messages from the group, can already get the message sender information
  + Added WhatsApp web and Chrome version information at initialization

**v2.1.0 (2021-07-30)**

* New features:
  + Added ReplyMessage (quoted message) feature
  + Added feature to send picture, audio, video and document messages via URL
  + Added a feature to save vcard messages in file form
  + Added ChangeState event to monitor connection status changes
* Improve:
  + Reading incoming messages with type locationand vcard(the data is already in the form of an object)

**v2.0.2 (2021-07-25)**

* Replaced the [Selenium Webdriver](https://www.selenium.dev/documentation/en/webdriver/) library with [the whatsapp-web.js](https://github.com/pedroslopez/whatsapp-web.js/) library as the main engine for communicating with WhatsApp Web.
* Full mode *siluman* (headless/no window)
* Not compatible with project examples that still use library version 1.3.x
* Grab groups and members to read group data and its members so you can save the results to the database
* Improved the function of sending messages with images, audio, video, and all types of documents
* Can save images, audio, video, and all types of documents from incoming messages (you can specify the storage location yourself)
* Can also read messages with type contactandlocation
* Archive chat
* Delete chat

**v1.3.3 (2021-02-02)**

* Minor update

**v1.3.2 (2020-11-22)**

* Bug fixes:
  + Send messages containing *emoji* in *stealth* mode (headless/no window).
* Improve:
  + The listen/subscribe method for incoming messages.
  + Method of sending message

**v1.3.1 (2020-07-20)**

* Bug fixes:
  + Sending a picture message to a contact that is not registered in WA contacts, the message sent becomes a duplicate. Example: Message **test** , sent as **testtest** .
* Improve:
  + Mode *siluman* (headless/no window).

**v1.3.0 (2020-07-17)**

* New features:
  + The subscribe message feature is sent, with this feature we can check whether the message has been sent or not.
* Bug fixes:
  + Sending a message to a contact that is not registered in WA contacts, the message sent becomes a duplicate. Example: Message **test** , sent as **testtest** .

**v1.2.0 (2020-07-13)**

* New features:
  + Added *stealth* mode (headless/no window)
  + Added grab contacts feature
* Improve:
  + Method SendMessageAsync, BroadcastMessage dan ListenIncomingMessageAsync

**v1.1.0 (2020-07-05)**

* New features:
  + Added a feature to send messages with pictures
  + Added a feature to send document files
* Improve:
  + Added support for .NET Framework 4.0
  + Incoming message subscribe feature
* Bug fixes:
  + Sending a contact with an invalid number
  + Sending multi-line messages during broadcast
* Deprecated:
  + Property  it's no longer used, so just disable it in the old code
  + Method no longer used, please use the *SendMessage* method instead

**v1.0.5 (2020-06-24)**

* Bug fixes:
  + Sending multi-line messages
* Improve:
  + Incoming message subscribe feature

**v1.0.0 (2020-06-22)**

* First release
* Sending message
* Send multiple messages (broadcast)
* Listen/subscribe incoming messages, so no need to add a timer object again to read incoming messages
* Auto replay of incoming messages
* Automatic login (So no need to scan the WA QRCode back and forth, just once)
* All existing methods except the method for connecting to the WA/Chrome browser, are made async (asynchronously) so they don't block the UI when executed.
* Support all types of databases

© 2020 - 2021 Kamarudin. Powered by [Jekyll](https://jekyllrb.com/) & [The Slate theme](https://github.com/pages-themes/slate/)  
[Coding4ever's Blog](http://coding4ever.net/) | [rudi.krsoftware@gmail.com](mailto:rudi.krsoftware@gmail.com)

Published with ❤️ by [GitHub Pages](https://pages.github.com/)